

Disclosure Packet

MS#: 302202.1

Title: Support for Dimension Attributes, and Multiple Hierarchies per Dimension, in OnLine Analytical Processing (OLAP)

Microsoft Team: Patricia Bornes, Kelly Carpenter

Inventors (names and email aliases):

Cristian Petculescu (cristp@microsoft.com);

Amir Netz (amirn@microsoft.com);

Mosha Pasumansky (moshap@microsoft.com);

Marius Dumitru (mariusd@microsoft.com);

Sasha Berger (sashab@microsoft.com);

Paul Sanders (paulsa@microsoft.com)

Component/Subcomponent: Data Warehouse / Analysis Services / Picasso

Summary: This invention defines a data model for storing data in relational database for access by OLAP. A dimension is defined for relational data that consists of a set of attributes. Each attribute is bound to a column in the relational database. A logical structure between the attributes can be defined, indicating how they are related. The attributes, and how they are related, define the full set of details of the dimension and their constraints. A dimension additionally may have a set of hierarchies defined for it. Each hierarchy is a sequence of attributes, presenting a common drill-down path that a user may follow. Each attribute may also be exposed as a simple hierarchy, containing just a single level of the attribute itself.

REDACTED

REDACTED

Microsoft Patent Predisclosure Document

Title of Invention: Support for dimension attributes, and multiple hierarchies per dimension, in OnLine Analytical Processing (OLAP).

Date: 18th Nov 2002

Document Author(s): Paul Sanders

Prior Disclosure

[Has there been any disclosure of the invention outside of Microsoft? If so, please identify the party (or parties) to whom disclosed, as well as the date and circumstances under which the disclosure was made (signed/unsigned non-disclosure agreement, etc.). Disclosure may include such things as an offer for sale, a demonstration, or a publication describing a novel aspect of the invention.]

Have been presentations & demonstrations to various MS partners and industry analysts, under NDA.

Introduction

[Please provide a high level description of the invention, including the names of the people who contributed to the invention.]

The invention is the extension of OLAP technology in order to:

- Allow a dimension to be defined in terms of the set of attributes that constitute the dimension.
- Allow multiple hierarchies to be defined per dimension, each hierarchy consisting of a sequence of attributes.

Contributors to the invention were:

Cristian Petculescu (Microsoft)
Amir Netz (Microsoft)
Mosha Pasumansky (Microsoft)
Marius Dumitru (Microsoft)
Sasha Berger (Microsoft)
Paul Sanders (Microsoft)

Strategic Importance of Invention:

[Please provide reasons why you think patent protection for this invention is important to Microsoft. Factors to consider include (1) is it core technology; (2) is it a feature that gives Microsoft a competitive advantage; (3) is it a feature that our competitors would want to copy; (4) does it include new APIs, file formats, network protocols, data schema or other components relating to product interoperability (5) is it related to a standard. Please include who you consider the most likely competitors and/or competitive products for this technology.]

The invention is of value as:

- It provides a richer data model, allowing business entities to be represented without much of the repetition and inefficiencies of today's technology.
- It increases the range of situations in which Microsoft OLAP technology can be used. It becomes more feasible and easier for an OLAP model to expose all of the columns of a relational dimension table, making it easier to use OLAP as the basis for all reporting, without direct access to the relational database at all.

REDACTED

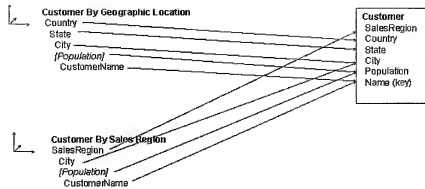
REDACTED

Motivation for the Invention:

[Describe (1) the problem addressed by the invention (e.g., limitations of prior products of Microsoft, or others), and (2) your solution to the problem (including what "new" things your invention does and a high-level description of how it does them).]

Consider the following example of the dimensions defined using the current MS OLAP technology. The relational database for Customer contains six columns (as shown in the diagram below). Logically, Customers can be organized either by their geographical location (Customers are in Cities, Cities are in State, and States are in Countries) or by sales region (Customers are in Cities, and Cities are assigned to Sales Regions – a sales region may contain cities from >1 country and a country maybe split into >1 sales region). In both cases, Cities have a population.

Using the current MS OLAP technology, this would result in two separate dimensions being defined, corresponding to the two different hierarchies of customer. The dimensions, their levels, and the mapping of the levels to the columns of the customer table, are shown below. In both the hierarchies, the City level has a 'member property' of Population (shown in *italics*), meaning that for any City, the Population can also be retrieved.



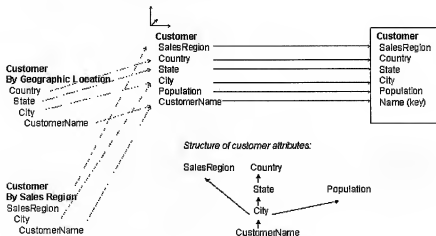
The problems with this are:

- There is much repetition of definition. The levels of CustomerName, City, and the member property Population, exist in both hierarchies, and their details must be repeated (e.g. their bindings to the columns of the customer table, their datatypes & names, and any custom rollup formulas).
- Along with the repetition of definition, comes inefficiencies due to the duplicate storage of data, and the need to retrieve the same details multiple times from the relational database.
- The client is limited to querying only the hierarchies defined by the cube designer. Hence in this example, a client could not request to see details by Country, broken down by the SalesRegions covering that Country. It becomes impractical for the designer of the cube to define hierarchies covering every possible drill-down path that the client might want. In addition, if the user were to request Country and SalesRegion, *all* combinations of Country and SalesRegion would be returned, even if no customers existed for any given combination.
- The separation between levels and member properties means that different mechanisms are used to retrieve them, even though to the client they are both 'attributes' of customer. In addition, the same item might be a level in one hierarchy, and a member property in another e.g. a different hierarchy in this example could have levels Population -> CustomerName.

This invention addressed these problems by extending the data model, such that:

- A dimension consists of a set of attributes. Each attribute is bound to a column in the relational database. The logical structure between the attributes can be defined, indicating how they are related e.g. every Customer is in one City, and every City is in one State, has one SalesRegion assigned, and has a Population. The attributes, and how they are related, defines the full set of details of the dimension and their constraints.
- A dimension additionally can have a set of hierarchies defined for it. Each hierarchy is a sequence of attributes, presenting a common drill-down path that the user will follow. Whilst some hierarchies present the 'natural' relationships between the data (e.g. Country -> State -> City -> CustomerName), they need not (e.g. a hierarchy SalesRegion -> Country could be defined).
- Each attribute can (optionally) be easily exposed as a simple hierarchy, containing just a single level of the attribute itself (plus, optionally, a level for 'All'). For example, the attribute Population could be exposed as a hierarchy with level 'All Populations', and a level whose members consist of every distinct value of the population column. Queries involving more than one such hierarchy (e.g. Country and SalesRegion) will only ever be return combinations for which there is at least one dimension member.

The equivalent model in this example would be as follows:



This solves the problems listed above as:

- There is no repetition of definition, nor of storage.

- All of the underlying columns can feasibly now be exposed as different hierarchies, in a uniform manner. Any combination of hierarchies can be combined by the client in the query, providing the same degree of flexibility as for a client querying the underlying table directly.

Description of the Invention:

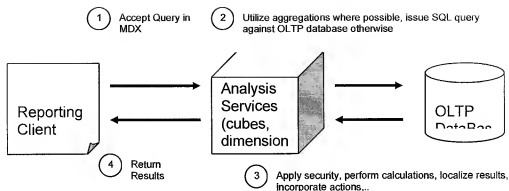
[Describe your proposed implementation of the invention, including the architecture and design details of the implementation. The design details should include a description of the component parts of, and individual operations performed by, your implementation. The use of a specific example, showing how the invention solves the problem being addressed, can be particularly helpful. You should also mention whether you have thought of any other implementations, or applications of, your invention. In most cases, 1-2 pages of description should be adequate to start the patent application process, although a more detailed description may greatly enhance the efficiency of the process.]

In addition to the summary information presented above:

- Adding additional hierarchies to a dimension need have little or any impact on the storage required. The performance decisions are made by the definition of the aggregates that should be pre-calculated and cached. These aggregates are defined in terms of the attributes (e.g. the designer can elect to maintain an aggregate by City, irrespective of the hierarchies defined)..
- Whilst the user sees hierarchies, and can utilize different hierarchies in a query, the results, and the aggregates that will be exploited to improve query performance, depend only on the 'coordinates' defined in terms of the members. E.g requesting the total sales of City 'Seattle' will be the same irrespective of whether the 'Customers By Country' or 'Customers By Sales Region' hierarchy were used. The cube designer therefore adds additional hierarchies just to ease the formulation of queries based upon the common hierarchies in which clients view the data.
- The separation into different attributes allows performance improvements during processing, where different attributes within a single dimension can be processed in parallel.
- The definition of the structure of the attributes allows the data to be validated during processing (e.g. to report an error if the same City appears in multiple Sttes).

Diagrams and Flow Charts:

[To support the description provided above, please include: (a) at least one block diagram showing the architecture of the system that implements your invention, and (b) at least one diagram illustrating the primary steps performed by your invention.]



Additional Information:

- List the names of any people who contributed to the invention.

As above.

- List any earlier, current or anticipated MS products that may use your invention:

SQL Server Analysis Services (Yukon release).

- List and attach (or provide pointers to) any documents that provide additional information about your invention or the product to which it relates, including specifications, journal articles, slide presentations, test/performance results, etc.]

Specification of the metadata that describes cubes/dimensions:

<http://msolap/specs/SMF/Picasso%20DDL%20and%20metadata.doc>

Specification of Dimensions:

<http://msolap/specs/Engines%20DM%20OLAP/Yukon%20dimension%20Architecture%20specification.doc>

- List any other sources that would provide helpful background information or illustrate prior work of others in this area (including, e.g., journal articles, text books, product literature, products, and specifications):

MSDN material for Analysis Services 2000 (current product)

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/olapdmad/aggettingstart_80xi.asp